AMENDMENTS TO THE CLAIMS

1. (Currently amended) A method for ~~providing a legal sequential combination of~~ ordered commands ~~for~~ verification testing of a multiprocessor computer system comprising multiple parallel processor threads, comprising ~~the steps of~~:

~~(a) providing a plurality of executable test commands;~~

~~(b)~~ providing at least one rule for forming a legal ordered sequences of semaphore test commands;

~~(c)~~ forming a plurality of ~~first~~ buckets of semaphore commands each comprising ~~at least two of~~ a plurality of the commands arranged in different ~~a first bucket~~ sequentially ordered command sequences legal under the at least one rule;

~~(d) forming a second bucket of commands comprising at least two of the commands arranged in a second bucket sequentially ordered sequence legal under the at least one rule; and~~

~~(e)~~ randomly selecting, ordering and combining ~~the~~ a plurality of the ~~first~~ buckets ~~and then the second bucket~~ in a sequential bucket test combination having a test sequential order, the sequential bucket test combination having [[a]] an ordered composite test command sequence, wherein an order of any of the buckets relative to another of the buckets within the test sequential order may be changed within the test sequential order, the test sequential order legal under the at least one rule and thereby generating a predictable result when executed by the threads; ~~wherein the second bucket and then the first bucket and may be combined into an alternate sequential bucket test combination having an alternate test sequential order, the alternate sequential bucket test~~

~~combination having an alternate composite test command sequence legal under the at least one rule~~

a semaphore manager sequentially distributing at least one each of the composite test sequence commands to a first plurality of the processor threads in a first thread distribution; and

observing the performance of each of the first plurality of threads in response to the distributed test commands.

2. (Currently amended) The method of claim 1, wherein the executable semaphore test commands may be selected from the group comprising ordered arguments of the computer system real operational code, ordered specific test instructions targeted for verification purposes, and randomly generated ordered instructions.

3. (Currently amended) The method of claim [[1]] 2, further comprising:

including a wait command in ~~wherein~~ the ~~executable~~ composite test command sequence test commands;

~~further comprise a~~ the wait command ~~configured to cause the computer system~~ causing the semaphore manager to pause for at least one instruction cycle~~, and wherein the step (c) of forming a first bucket of commands further comprises the following step: (i) including said wait command in said first bucket~~ and skip a next first plurality thread and thereby distribute at least one next command ordered subsequent to the wait command to an alternative thread, the manager thereby sequentially distributing the test

combination commands to each of a second plurality of the processor threads different from the first plurality; and

observing the performance of each of the second plurality of threads in response to the distributed test commands.

4. (Currently amended) The method of claim [[1]] 3 wherein ~~the step (e)~~ combining the ~~first and second~~ buckets further comprises ~~the following step: (j)~~ including said wait command within a bucket, between ~~the~~ first and second buckets, before the first bucket, or after the second bucket.

5. (Currently amended) The method of claim 3 ~~wherein the step (i) of including said wait command in said first bucket~~ further ~~comprises~~ comprising the step of: ~~(m)~~ randomly ~~selecting a point of insertion~~ inserting the wait command within the composite test command ~~bucket sequentially ordered~~ sequence.

6. (Cancelled)

7. (Currently amended) The method of claim 5 wherein ~~the step of (i) of~~ including said wait command ~~in said first bucket further~~ comprises ~~the step of (o)~~ inserting a random amount of the wait command.

8. (Cancelled)

9. (Currently amended) The method of claim 7, ~~1 wherein the step (a) of providing a plurality of executable test commands~~ further comprising ~~comprises the steps of~~:

~~(q)~~ providing a parameter for at least one of the plurality of executable commands; ~~and~~

~~(r)~~ assigning a value to the parameter; and

identifying and selecting a next thread for a next at least one command in response to the parameter value.

10. (Cancelled)

11. (Currently amended) The method of claim 9 ~~wherein the step (r) of assigning a value to the parameter is performed~~ further comprising randomly selecting and unlocking a semaphore in response to the parameter value.

12. (Currently amended) A verification testing system for a multiprocessor computer system comprising multiple parallel processor threads, comprising:

~~(a) a microprocessor;~~

~~(b)~~ a ~~central~~ semaphore manager connected to the microprocessor and configured to rout ordered semaphore command instructions to the microprocessor threads; and

~~(c) a plurality of executable test commands;~~

~~(d)~~ a memory device comprising a plurality of buckets of executable ordered semaphore test commands, each bucket comprising a plurality of the commands arranged in different sequentially ordered command sequences legal under at least one rule for forming a legal ordered sequences of semaphore commands;

~~(e) a first bucket of commands comprising at least two of the executable test commands in a first bucket sequentially ordered sequence legal under the at least one rule; and~~

~~(f) a second bucket of commands comprising at least two of the commands arranged in a second bucket sequentially ordered sequence legal under the at least one rule;~~

~~wherein the central manager is configured to combine the first bucket and then the second bucket in a sequential bucket test combination having a test sequential order, the sequential bucket test combination having a composite test command sequence legal under the at least one rule, and run the sequential bucket test combination on the microprocessor; and~~

~~wherein the central manager is further configured to combine the second bucket and then the first bucket into an alternate sequential bucket test combination having an alternate test sequential order, the alternate sequential bucket test combination having an alternate composite test command sequence legal under the at least one rule, and run the alternate sequential bucket test combination on the microprocessor~~

the semaphore manager configured to combine buckets by randomly selecting the buckets and an order of the buckets within a sequential bucket test combination wherein an order of any of the buckets relative to another of the buckets may be changed and the

sequential test combination remain legal under the at least one rule and thereby generate a predictable result when executed by the threads; and

wherein the semaphore manager is configured to sequentially distribute at least one each of the composite test command sequence commands to a first plurality of the processor threads and observe the performance of each of the first plurality of threads.

13. (Currently amended) The system of claim 12, wherein the executable ordered semaphore test commands ~~may be~~ are selected from the group comprising ordered arguments of the computer system real operational code, ordered specific test instructions targeted for verification purposes, and randomly generated ordered instructions.

14. (Currently amended) The system of claim ~~12~~ 13, wherein the ~~executable test commands~~ sequential bucket test combination further comprises a wait command, and wherein the semaphore manager is configured to ~~cause the computer system to~~ pause for at least one instruction cycle in response to the wait command~~, and wherein the first bucket of commands further comprises said wait command~~ and skip a next first plurality thread and distribute at least one next command ordered subsequent to the wait command to an alternative thread, the manager thereby sequentially distributing the test combination commands to each of a second plurality of the processor threads different from the first plurality; and

observe the performance of each of the second plurality of threads in response to the distributed test commands.

15. (Currently amended) The system of claim ~~12~~ 14, ~~wherein the executable test commands further comprise a wait command configured to cause the computer system to pause for at least one instruction cycle,~~ wherein sequential bucket test combination further comprises said wait command within a bucket, between ~~the~~ first and second buckets, before the first bucket, or after the second bucket.

16. (Currently amended) The system of claim ~~12~~ 15, wherein the executable test commands further comprise a parameter for at least one of the plurality of executable commands, and wherein the semaphore ~~central~~ manager is configured to assign a value to the parameter and to identify and select a next thread for a next at least one command in response to the parameter value.

17. (Cancelled)

18. (Currently amended) The system of claim 16 wherein the semaphore ~~central~~ manager is configured to randomly ~~assign a value to the parameter~~ select and unlock a semaphore in response to the parameter value.

19. (Cancelled).

20. (New) An article of manufacture comprising a computer readable medium having a computer readable microprocessor semaphore manager program, wherein the semaphore manager program, when executed on a computer, causes the computer to:

form a plurality of buckets of ordered semaphore test commands each comprising a plurality of commands arranged in different sequentially ordered command sequences legal under at least one rule for forming a legal ordered sequence of semaphore test commands;

randomly select, order and combine a plurality of the buckets in a sequential bucket test combination having a test sequential order, the sequential bucket test combination having an ordered composite test command sequence, wherein an order of any of the buckets relative to another of the buckets within the test sequential order may be changed within the test sequential order, the test sequential order legal under the at least one rule, and thereby generate a predictable result when executed by the threads;

sequentially distribute at least one each of the composite test sequence commands to a first plurality of multiprocessor computer system parallel processor threads in a first thread distribution; and

observe the performance of each of the first plurality of threads in response to the distributed test commands.

21. (New) The article of manufacture of claim 20, wherein the ordered test commands are selected from the group comprising ordered semaphore arguments of computer system real operational code, specific ordered semaphore test instructions targeted for verification purposes, and randomly generated ordered semaphore instructions, and the composite test command sequence further includes a wait command, and wherein the semaphore manager program, when executed on a computer, causes the computer to:

pause for at least one instruction cycle and skip a next first plurality thread and distribute at least one next command ordered subsequent to the wait command to an alternative thread, the computer thereby sequentially distributing the test combination commands to each of a second plurality of the processor threads different from the first plurality; and

observe the performance of each of the second plurality of threads in response to the distributed test commands.

22. (New) The article of manufacture of claim 21, wherein the semaphore manager program, when executed on a computer, causes the computer to randomly insert the wait within a bucket, between first and second buckets, before the first bucket, or after the second bucket.

23. (New) The article of manufacture of claim 22, wherein the semaphore manager program, when executed on a computer, causes the computer to insert a random amount of the wait command.

24. (New) The article of manufacture of claim 23, wherein the semaphore manager program, when executed on a computer, causes the computer to assign a value to a parameter within at least one of the plurality of executable commands, and identify and select a next thread for a next at least one command in response to the parameter value.

25. (New) The article of manufacture of claim 24, wherein the semaphore manager program, when executed on a computer, causes the computer to randomly select and unlock a semaphore in response to the parameter value.